

## Trabajo Fin de Grado

### AUTÓMATAS PROGRAMABLES CONECTADOS A LA RED ELÉCTRICA POR POWER LINE COMMUNICATION (PLC)

PROGRAMMABLE AUTOMATA CONNECTED BY POWER LINE  
COMMUNICATION (PLC)

Autor:

Guillermo Villafaña Samper

Director:

Alfredo Sanz Molina

Grado de Ingeniería Electrónica y Automática

ESCUELA DE INGENIERÍA Y ARQUITECTURA

Junio 2021

## **AUTÓMATAS PROGRAMABLES CONECTADOS A LA RED ELÉCTRICA POR POWER LINE COMMUNICATION (PLC)**

**(en inglés) PROGRAMMABLE AUTOMATA CONNECTED BY POWER LINE COMMUNICATION (PLC)**

### **RESUMEN**

---

Desde hace ya unas cuantas décadas, la utilización de autómatas programables en la industria se ha convertido en una práctica habitual para muchas compañías gracias a su eficacia en la reducción tanto de costes como de tiempos de producción. Sin embargo, su implantación en instalaciones de cierto tamaño supone complicaciones de conexión entre ellos mismos y también con sus numerosas entradas y salidas.

Lo que se plantea en este proyecto como meta final es utilizar la tecnología de comunicación por la red eléctrica (Power Line Communications) para que un autómata programable pueda utilizar diversos sus módulos de entradas y salidas distribuidos por la instalación a controlar, que utilice la red eléctrica de la instalación como medio de comunicación o bus de campo. Finalmente, la investigación ha llevado a desarrollar una “Prueba de concepto” de la propuesta de producto inicial, en la que se consigue compartir entradas y salidas de otros módulos a través de la red para la ejecución de un autómata programable desde un ordenador, cuyo programa puede estar diseñado en cualquiera de los cinco lenguajes de autómata normalizados.

Todo esto conseguido sobre un montaje simple, ya que el cableado empleado es mínimo aunque aumente el número de módulos, lo que proporciona una ventaja (la flexibilidad) respecto a las comunicaciones industriales convencionales.

Quiero destacar también que este trabajo ha sido posible gracias al soporte de la empresa Microchip Technology y en especial de su grupo de Smart Energy dirigido en España por Alfredo Sanz, lo que me ha permitido disponer del material necesario para poder superar tanto los objetivos iniciales como los distintos problemas que han ido surgiendo.



Escuela de  
Ingeniería y Arquitectura  
Universidad Zaragoza

## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe entregarse en la Secretaría de la EINA, dentro del plazo de depósito del TFG/TFM para su evaluación).

D./D<sup>ña</sup>. GUILLERMO VILLAFANÍA SAMPER ,en

aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)

Grado

(Título del Trabajo)

AUTÓMATAS PROGRAMABLES CONECTADOS A LA RED ELÉCTRICA POR POWER LINE COMMUNICATIONS (PLC) (en inglés, PROGRAMMABLE AUTOMATA CONNECTED BY POWER LINE COMMUNICATIONS (PLC))

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, a 21 de junio de 2021

Fdo: GUILLERMO VILLAFANÍA SAMPER

# ÍNDICE

---

## ÍNDICE GENERAL

<b>RESUMEN .....</b>	<b>1</b>
<b>ÍNDICE .....</b>	<b>3</b>
ÍNDICE GENERAL.....	3
ÍNDICE DE FIGURAS .....	4
ÍNDICE DE TABLAS.....	4
TABLA DE ABREVIATURAS .....	4
<b>1 INTRODUCCIÓN .....</b>	<b>6</b>
1.1 OBJETIVOS Y ALCANCE DEL PROYECTO.....	6
1.2 PROBLEMAS PROPUESTOS .....	6
1.3 METODOLOGÍA DE TRABAJO .....	7
1.4 ETAPAS PLANTEADAS .....	8
1.5 CONTENIDO DE LA MEMORIA .....	8
<b>2 FUNDAMENTOS TEÓRICOS.....</b>	<b>10</b>
2.1 AUTÓMATAS PROGRAMABLES.....	10
2.1.1 <i>Lenguaje LD</i> .....	11
2.1.2 <i>Lenguaje FBD</i> .....	12
2.1.3 <i>Lenguaje IL</i> .....	12
2.1.4 <i>Lenguaje ST</i> .....	12
2.1.5 <i>Lenguaje SFC</i> .....	13
2.2 OPENPLC.....	13
2.2.1 <i>Editor</i> .....	14
2.2.2 <i>Runtime</i> .....	14
2.2.3 <i>Alternativas a OpenPLC</i> .....	15
2.3 POWER LINE COMMUNICATIONS (PLC).....	15
2.3.1 <i>Normativas y frecuencias</i> .....	16
2.3.1.1 Estándar G3-PLC.....	16
2.3.1.2 Estándar PRIME .....	16
2.3.1.3 Capas de comunicación.....	16
2.3.1.4 Bandas de frecuencias.....	17
2.3.2 <i>Hardware/dispositivos</i> .....	18
2.3.2.1 Kit de evaluación PL360G55CB-EK.....	18
2.3.2.2 Dispositivo PL360 .....	19
<b>3 DESARROLLO DEL PROYECTO .....</b>	<b>20</b>
3.1 PLANTEAMIENTO INICIAL .....	20
3.2 DIFICULTADES ENCONTRADAS .....	21
3.3 SOLUCIÓN ADOPTADA .....	22
3.3.1 <i>Aplicación ‘PLC and Go’</i> .....	22
3.3.2 <i>Inserción de una “placa de control”</i> .....	23
3.3.3 <i>Placa base y placas servicio de la red</i> .....	24
3.3.3.1 Placa base.....	24
3.3.4 <i>Análisis de los resultados obtenidos</i> .....	25
<b>4 CONCLUSIONES.....</b>	<b>28</b>
<b>5 BIBLIOGRAFÍA .....</b>	<b>29</b>

<b>6</b>	<b>ANEXOS .....</b>	<b>33</b>
6.1	ANEXO A: MANUAL DE USO DEL SISTEMA .....	33
6.2	ANEXO B: CARPETA CON LOS PROYECTOS SOFTWARE Y LAS APLICACIONES BINARIAS .....	33

## ÍNDICE DE FIGURAS

FIGURA 1:	HARDWARE GENERAL DE UN AP .....	10
FIGURA 2:	MODO DE OPERACIÓN DE UN AP .....	11
FIGURA 3:	DESCRIPCIÓN DE ELEMENTOS DEL LENGUAJE LD .....	11
FIGURA 4:	EJEMPLO DEL LENGUAJE LD .....	12
FIGURA 5:	EJEMPLO DEL LENGUAJE FBD .....	12
FIGURA 6:	INSTRUCCIONES BÁSICAS Y EJEMPLOS DEL LENGUAJE IL .....	12
FIGURA 7:	EJEMPLO DEL LENGUAJE ST .....	13
FIGURA 8:	MODELO GENERAL Y EJEMPLO CON LENGUAJE SFC .....	13
FIGURA 9:	EJEMPLO DE UN ARCHIVO DE FORMATO .ST .....	14
FIGURA 10:	ESTRUCTURA DE LAS CAPAS DE COMUNICACIÓN SEGÚN EL PROTOCOLO ESCOGIDO .....	17
FIGURA 11:	BANDAS DE FRECUENCIAS NORMALIZADAS PARA PLC .....	18
FIGURA 12:	VISTA CENITAL DEL KIT PL360G55CB-EK .....	18
FIGURA 13:	EJEMPLO DEL PROGRAMA TERATERM .....	20
FIGURA 14:	VISTA GENERAL DE LA INTERFAZ DE ATMEL STUDIO .....	21
FIGURA 15:	REPRESENTACIÓN DE LA COMUNICACIÓN PLC "MULTI-PUNTO" .....	23
FIGURA 16:	VISTA DE UNA PLACA SAMD21 XPLAINED PRO .....	24
FIGURA 17:	VISTA CENITAL DE LA ESTRUCTURA FINAL .....	25

## ÍNDICE DE TABLAS

TABLA 1:	RESUMEN ABREVIATURAS EMPLEADAS .....	4
----------	--------------------------------------	---

## TABLA DE ABREVIATURAS

*Tabla 1: Resumen abreviaturas empleadas*

Abreviatura	Nombre completo
AC	Tensión alterna.
ADP	Adaptation layer (en español, "capa de adaptación").
AP	Autómata Programable. Para acortar este nombre no se utiliza la abreviatura más común dentro del mundo de la automatización ya que coincidiría con una ya empleada: PLC, Programmable Logic Controller (en español, "controlador lógico programable").
API	Application Programming Interface (en español, "interfaz de programación de aplicaciones").
CENELEC	European Committee for Electrotechnical Standardization (en español, "Comité Europeo de Normalización Electrotécnica").
CPU	Central Processing Unit (en español, "Unidad Central de Procesamiento").
DC	Tensión continua.
FBD	Function Block Diagram (en español, "Diagrama de Bloques de Función").
GPL	General Public License (en español, "Licencia General Pública").
HMI	Human Machine Interface (en español, "Interfaz Humano Máquina").
I/O	Inputs/Outputs (en español, "Entradas/Salidas").

IEC	International Electrotechnical Commission (en español, “Comisión Electrotécnica Internacional”).
IEEE	Institute of Electrical and Electronics Engineers (en español, “Instituto de Ingenieros Eléctricos y Electrónicos”).
IL	Instruction List (en español, “Lista de Instrucciones”).
ITU	International Telecommunications Union (en español, “Unión Internacional de Telecomunicaciones”).
LD	Ladder Diagram (en español, “lenguaje de contactos”).
LED	Light Emitting Diode (en español, “diodo emisor de luz”).
MAC	Medium Access Control layer (en español, “capa de control de acceso al medio”).
MAC RT	MAC Real Time layer (en español, MAC “en tiempo real”).
MÓDEM	MODulador DEModulador.
NA	Normalmente Abierto
NC	Normalmente Cerrado
PC	Personal Computer (en español, “ordenador personal”).
PCB	Printed Circuit Board (en español, “placa de circuito impreso”).
PHY	Physical layer (en español, “capa física”).
PLC	Power Line Communications (en español, “comunicaciones mediante línea de potencia”).
PRIME	PoweRline Intelligent Metering Evolution (en español, “evolución de la medición inteligente por línea eléctrica”).
RE	Red Eléctrica.
SFC	Sequential Function Chart (en español, “gráfico secuencial de funciones”).
ST	Structured Text (en español, “Texto estructurado”).
TFG	Trabajo Fin de Grado.
uC	Microcontrolador. La u simboliza a la letra griega ‘mu’ que en los sistemas de numeración representa el orden de magnitud ‘micro’ ( $10^{-6}$ ).
USART	Universal Synchronous/Asynchronous Receiver-Transmitter (en español, “Transmisor-Receptor Síncrono/Asíncrono Universal”).
USB	Universal Serial Bus (en español, “Bus Serie Universal”)

# 1 INTRODUCCIÓN

---

## 1.1 OBJETIVOS Y ALCANCE DEL PROYECTO

El objetivo principal de este proyecto es construir un sistema compuesto por una serie de Autómatas Programables (a partir de aquí, AP) comunicados a través de la red eléctrica (a partir de aquí, RE) mediante la tecnología Power Line Communications (a partir de aquí, PLC) para conseguir expandir sus módulos de entradas y salidas (a partir de aquí, I/O).

La utilidad para el mundo industrial de esta propuesta es que facilita las conexiones entre los diversos módulos de I/O de los AP, en especial para aplicaciones en los entornos industriales de medias y grandes dimensiones. En los últimos años, la flexibilidad y ahorro considerable en tiempo y dinero demostrados por los AP en el mundo industrial los han convertido en un componente esencial en la mayoría de los procesos productivos. Sin embargo, un inconveniente para las compañías se lo encuentran en sus fábricas cuando se ven forzadas a un cableado demasiado complejo (buses de comunicación industrial) para conectar sus AP a sus módulos de I/O correspondientes, que pueden estar alejados en distintos sitios de la misma cadena de montaje, o también para intercomunicar varios autómatas entre ellos.

Los buses de comunicación industrial son el enlace más común entre las I/O y los sistemas de control en un proceso industrial (en este caso los autómatas). También llamados “buses de campo”, pueden transmitir varias señales de datos simultáneamente y, como he dicho ya, a nivel económico no salen muy rentables por la abundante longitud de cable que requieren. [\[1\]](#)

La novedad de este TFG es el uso de la tecnología PLC para la conexión distribuida de los módulos de las I/O de cada AP con la intención de lograr que todos los AP puedan tener un funcionamiento sincronizado. Además, el programa de autómata que estén ejecutando será el mismo en todos los dispositivos, de forma que la variación de entradas o salidas en uno de los elementos de esta red pueda influir en salidas de otro u otros que pertenezcan a la misma red.

La consecuencia práctica de alcanzar esta meta es la reducción drástica del cableado utilizado gracias a que la estructura sobre la que se sustenta todo este sistema va a ser sencilla, lo cual repercutirá favorablemente en la aplicación industrial que se ha explicado unos párrafos atrás.

Con este escenario, se abren varios retos que serán fundamentales para el desarrollo de este trabajo: programar los dispositivos, en este trabajo serán microcontroladores (a partir de aquí, uC), que se emplearán en el proceso para que funcionen y ejerzan como un AP; hallar una aplicación interactiva con la que crear programas de autómata propios; y hallar el método para intercambiar entre los AP la información de cada una de las I/O expandidas por la red.

## 1.2 PROBLEMAS PROPUESTOS

Para que un uC actúe como un AP y también el de poder crear por parte de la persona que maneje esta estructura un programa de autómata, se requiere de alguna herramienta externa, web o de aplicación a instalar, que se pueda controlar desde un ordenador (a partir de aquí, PC “Personal Computer”) con la que lograr todo esto. En la segunda cuestión, la situación hipotética sería que el propio usuario desde un PC pueda diseñar su programa de autómata deseado. Éste sería guiado a través del propio PC para primero configurar correctamente el sistema al completo y luego crear el programa que desee conociendo la cantidad total de registros (entradas y salidas sumadas de todos los AP) que dispone.

Tras la búsqueda correspondiente se encontraron varios proyectos que encajaban en los requisitos fundamentales: apariencia e interfaz sencillas, permitían ejecutar AP's en uC cargando sus desarrollos de código, disponían de aplicaciones de PC que permiten elaborar un programa de autómatas propio, y algo clave, eran proyectos de código libre y abierto. Finalmente se optó por una de las opciones principalmente porque era la que más soporte en distintas arquitecturas de uC tenía.

El seleccionado fue el proyecto bajo el nombre OpenPLC, creado por un grupo de investigadores liderados por el Doctor de la Universidad de Alabama Thiago Alves para poder implementar en la mayoría de los uC convencionales el comportamiento de un AP, consiguiendo así, con una arquitectura sencilla, donde desarrollar aplicaciones industriales, con lo que soluciona el primero de los dilemas mencionados al final del [apartado 1.1](#). [\[2\]](#) [\[3\]](#)

La característica de que sea un proyecto de código libre y abierto ("Open Source Project") es importante ya que el tipo de licencia de que disponga el proyecto marcará su uso y empleo para unas aplicaciones futuras, y para otras no (más en [capítulo 2.2](#)). En OpenPLC, como se puede observar en la referencia adjunta en la bibliografía, tiene actualmente una licencia GPL-v3.0 (GPL, las siglas de "General Public License"). [\[4\]](#)

Aunque la mayoría de las cuestiones afrontadas en este TFG eran conocidas por el autor de su etapa universitaria (en el grado de Ingeniería Electrónica y Automática se tratan mucho aspectos como la programación de uC, concepto de AP y cómo controlar su ejecución, ...), algunos otros temas demandaron un estudio previo para luego afrontarlos con un conocimiento básico de ese contenido, como fueron los tipos de comunicación que son necesarios en el proyecto: la tecnología PLC o la comunicación USART (significado de la abreviatura en [Tabla 1](#)). Para estos casos existe documentación abierta en la red ("datasheets" o fichas de datos, notas de aplicación, "user guides" o guías de usuario) para obtener un nivel de información suficiente y comprender estos elementos.

Como último problema surgido a la hora del planteamiento inicial del proyecto, será necesario armar un montaje a nivel eléctrico donde organizar y unir adecuadamente las conexiones entre los AP y sus módulos de I/O, tanto digitales como analógicos, que se quieran configurar para cada uC.

### 1.3 METODOLOGÍA DE TRABAJO

Con el tema, los objetivos y los problemas definidos para este TFG, en esta sección se expondrá la metodología seguida durante su elaboración para superarlos, separando en: análisis a nivel hardware (a nivel físico, el que se observa a simple vista del montaje) y análisis a nivel software (a nivel de aplicación, programación de código). [\[5\]](#)

Desde la perspectiva hardware, partiendo del objetivo de alcanzar de los AP en ejecución con sus módulos de I/O distribuidos por PLC, se requerirá de una cantidad considerable de placas de circuito impreso (a partir de aquí, PCB) con un uC que acepte el software de OpenPLC y con la tecnología PLC incorporada para poder intercomunicarla con el resto.

Otra cuestión importante en el aspecto físico del sistema es considerar cuál va a ser la estructura donde se situarán los módulos de I/O de los AP's, ya que tiene que permitir configuraciones eléctricas para poder conectar óptimamente las distintas entradas y salidas con sus correspondientes pines externos de los dispositivos principales; y también un grado de



flexibilidad en esas uniones porque se harán muchos cambios en ellas en la fase de pruebas y tests del proyecto.

Pasando a las herramientas del ámbito software, ya se ha introducido la aplicación OpenPLC en la sección anterior. Será fundamental tanto desde el PC para el diseño del programa de autómatas, como en el desarrollo que me proporcione para cargar en los uC y alcanzar el comportamiento de AP en estos dispositivos.

En este último aspecto de subir una aplicación a las PCB, se necesitará un programa que permita editar y compilar código y también que genere el archivo correspondiente a él compatible con la arquitectura del uC destino.

## 1.4 ETAPAS PLANTEADAS

Primero, se trabajó en investigar y experimentar sobre la ‘desconocida’ aplicación OpenPLC, para saber más sobre su actividad y ver cómo y dónde incluir las modificaciones a realizar para conseguir que funcione como se ha estipulado.

Una vez trabajado sobre este tema, se apartó temporalmente para comenzar a manejar por primera vez las PCB con tecnología PLC tanto mediante tests de comunicación sencillos entre varias de ellas como con programas en los que interactúe con sus pines externos.

En el siguiente paso se comenzó la implantación del software de AP sobre una de las PCB, asegurando que la adaptación se produce correctamente con OpenPLC. De momento sin mezclar con el software correspondiente a la comunicación por RE, aquí sólo se actuó sobre hacer funcionar un único controlador ejecutando un programa de autómatas con exclusivamente su módulo de I/O.

Finalmente, la última fase del trabajo se basó en llegar a la solución final, con un número de AP ejecutándose simultáneamente con sus módulos de I/O distribuidos por la red PLC creada e intercambiando información entre ellos. Esto permite a cada dispositivo con OpenPLC instalado actualizar el estado y los valores de los registros y, si corresponde según el programa, actuar sobre una o varias de sus salidas.

## 1.5 CONTENIDO DE LA MEMORIA

La organización que va a seguir este documento a partir de este punto se encuentra dividida en:

- Fundamentos teóricos ([capítulo 2](#)), donde se detallarán los conceptos dentro del campo teórico tratados durante el proceso de elaboración de este TFG.
- Desarrollo del proyecto ([sección 3](#)), la parte más extensa, donde al comienzo se realizará la presentación de cómo afrontar los objetivos fijados en esta [Introducción](#), aparecerán también las investigaciones y experimentos realizados para llegar a la solución final, y para acabar se analizará lo obtenido al finalizar el proceso.
- Conclusiones ([apartado 4](#)). Espacio destinado a resolver si tras el trabajo efectuado se ha logrado cumplir o no la hipótesis general planteada para ser aplicada en la sociedad actual. En este proyecto, sería demostrar si se ha conseguido finalmente distribuir los módulos de I/O de un AP por la red eléctrica y reducir así el cableado y el gasto empleados.
- Referencias bibliográficas ([5](#)).

- Anexos ([punto 6](#)) que en esta Memoria serán dos elementos: un “Manual de uso” del sistema construido; y un fichero que contendrá: los proyectos software desarrollados para examinar el código, y las aplicaciones correspondientes a ellos y que deberán ser cargadas en las placas para hacer funcionar el conjunto de forma óptima.

## 2 FUNDAMENTOS TEÓRICOS

### 2.1 AUTÓMATAS PROGRAMABLES

Los autómatas programables surgen de la necesidad sobre mitad del siglo XX de un sistema de control flexible y fácil de programar en entornos industriales para dejar atrás los circuitos secuenciales a base de relés. En éstos, la lógica de control era fija, y si el sistema de producción que manejaban cambiaba, se tenía que desmontar, rediseñar, y volver a conectar toda esa instalación con el coste que eso llevaba, a lo que se sumaba también un gasto de mantenimiento elevado de los propios relés.

El uso y cantidad de AP's aumentó, y con ello los protocolos para su diseño y programación, lo que llevó a publicar estándares a nivel internacional para unificar estos aspectos y alcanzar un convenio común dentro del mundo de la automatización industrial. La normativa vigente actualmente se revisó por última vez en 2003 por la Comisión Electrotécnica Internacional (IEC), se le denominó IEC 61131-3 y declara varios modos de programación normalizados. [6]

La estructura física de estos controladores programables se compone de (Figura 1): [6]

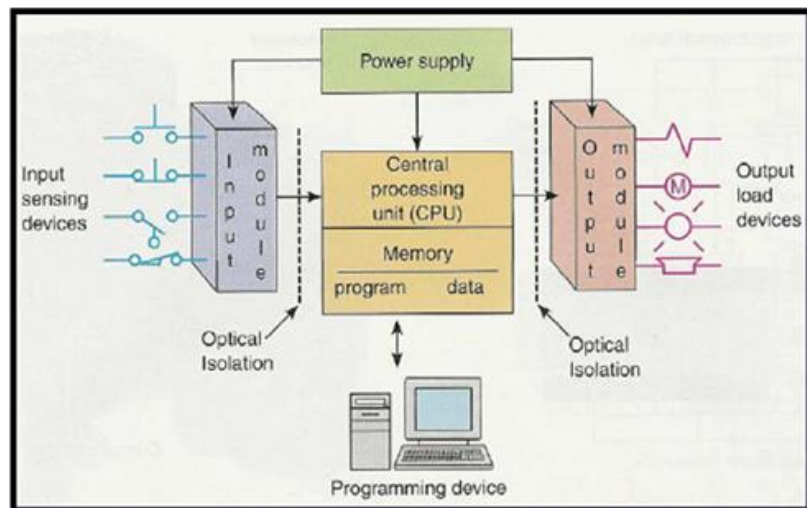


Figura 1: Hardware general de un AP

- Unidad de procesamiento central (a partir de aquí, CPU), componente imprescindible que lee las entradas, ejecuta el programa y maneja las distintas salidas.
- Fuente de alimentación ("Power supply"), aporta energía y tensión al sistema.
- Módulos de I/O. Comunican la CPU con los elementos de campo del sistema (I/O).
- Unidad de programación (PC, pantallas manuales o táctiles).

Volviendo a la CPU, sus procesos software internos serán los que marquen el funcionamiento del AP, basado en un bucle cíclico como el de la Figura 2. [7]

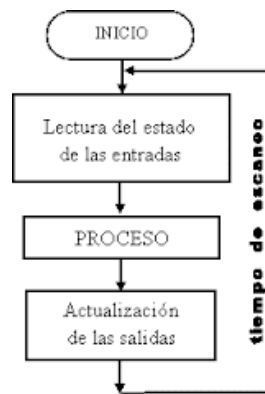


Figura 2: Modo de operación de un AP

Los programas de autómatas también pueden ser diseñados en diversos lenguajes, pero sólo algunos están reconocidos como válidos en el ámbito internacional. Acorde con el estándar IEC 61131-3 imperante, se distinguen cinco modos de programación de autómatas normalizados: en las siguientes subsecciones se añadirá una breve explicación de cada uno de ellos. Los significados de sus siglas identificativas se especifican en el [Tabla 1](#).

### 2.1.1 Lenguaje LD

También denominado “lenguaje de contactos” ya que fue el primero que se extendió al tener un diseño similar a los elementos que se encontraban en los anteriormente empleados armarios de contactos, se centra en un diagrama que maneja señales lógicas como componentes que controlan la conexión entre 2 líneas verticales de potencia. De éstos (que pueden ser variables, funciones ...) destacan los contactos y las bobinas, resumidos en la Figura 3 ([7]).

Símbolo	Nombre	Descripción
	Contacto NA	Se activa cuando hay un uno lógico en el elemento que representa, esto es, una entrada (para captar información del proceso a controlar), una variable interna o un bit de sistema.
	Contacto NC	Su función es similar al contacto NA anterior, pero en este caso se activa cuando hay un cero lógico, cosa que deberá de tenerse muy en cuenta a la hora de su utilización.
	Bobina NA	Se activa cuando la combinación que hay a su entrada (izquierda) da un uno lógico. Su activación equivale a decir que tiene un uno lógico. Suele representar elementos de salida, aunque a veces puede hacer el papel de variable interna.
	Bobina NC	Se activa cuando la combinación que hay a su entrada (izquierda) da un cero lógico. Su activación equivale a decir que tiene un cero lógico. Su comportamiento es complementario al de la bobina NA.
	Bobina SET	Una vez activa (puesta a 1) no se puede desactivar (puesta a 0) si no es por su correspondiente bobina en RESET. Sirve para memorizar bits y usada junto con la bina RESET dan una enorme potencia en la programación.
	Bobina SET	Permite desactivar una bobina SET previamente activada.

Figura 3: Descripción de elementos del lenguaje LD

En la Figura 4, el AP maneja cuatro señales lógicas: dos entradas digitales (“digInput”), contactos Normalmente Abiertos (NA); y dos salidas digitales (“digOutput”), bobinas también NA. Se activarán (salida a ‘1’) cuando les llegue el flujo de su línea vertical de potencia correspondiente: cuando el contacto de cada segmento horizontal se accione (la entrada se pone a ‘1’).

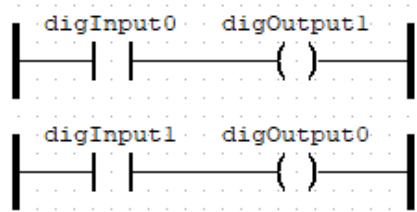


Figura 4: Ejemplo del lenguaje LD

### 2.1.2 Lenguaje FBD

Este lenguaje se apoya en una visualización gráfica por bloques con las operaciones matemáticas o lógicas que se ejecutan en el programa. En la Figura 6 se ilustra una función tipo AND con dos entradas y una salida digitales. [8]

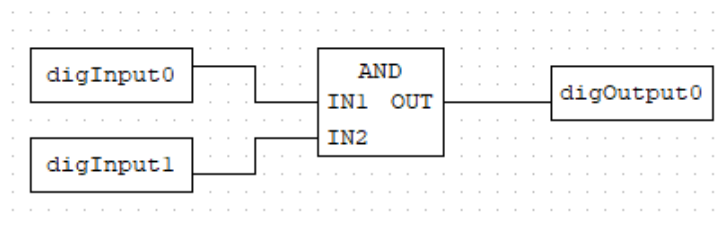


Figura 5: Ejemplo del lenguaje FBD

### 2.1.3 Lenguaje IL

Lenguaje textual de bajo nivel, es el que más ha caído en desuso. En la Figura 6 hay ejemplos de instrucciones sencillas comparándolas con programas LD equivalentes. [9] [10]

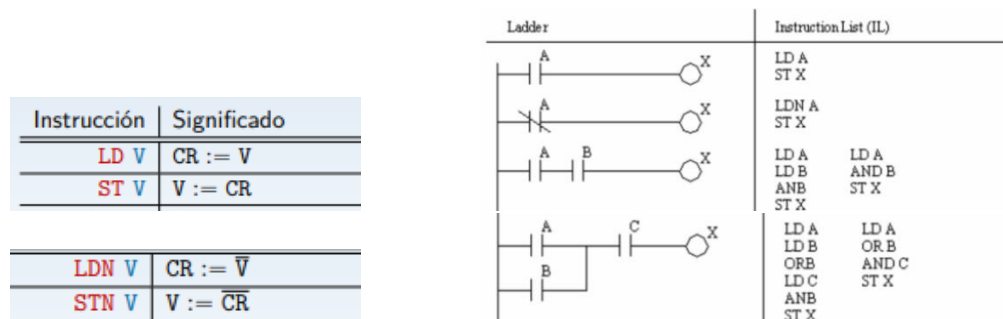


Figura 6: Instrucciones básicas y ejemplos del lenguaje IL

CR, Current Result (resultado actual), almacena los resultados de las instrucciones precedentes.

### 2.1.4 Lenguaje ST

Programación textual de alto nivel que recoge muchos aspectos de los lenguajes convencionales Pascal o 'C' y permite implementar estructuras iterativas o condicionales (ejemplo en Figura 7).

```

1 PROGRAM example
2 VAR
3     input1 : BOOL;
4     output1 : BOOL;
5 END_VAR
6
7 IF input1 = TRUE THEN
8     output1 := TRUE
9 ELSE
10    output1 := FALSE
11 END_IF
12 END_IF

```

Figura 7: Ejemplo del lenguaje ST

### 2.1.5 Lenguaje SFC

Representación gráfica similar a un diagrama de flujo que emplea etapas, transiciones y acciones como su antecesor: el estándar GRAFCET (GRAFica de Control de Etapas de Transición). El ejemplo que se adjunta es la implementación de una función lógica AND de dos entradas y una salida digitales. [7]

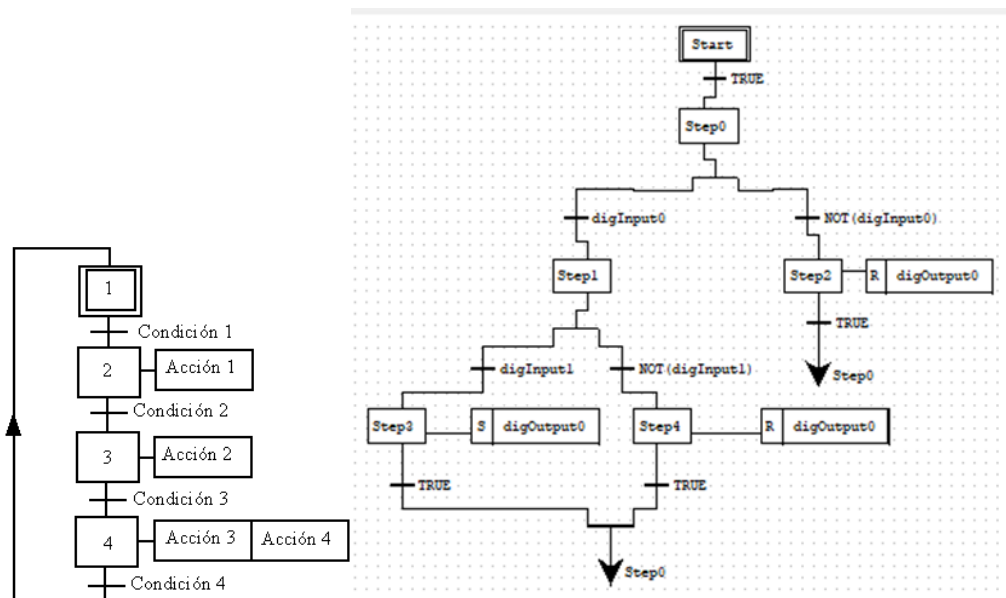


Figura 8: Modelo general y ejemplo con lenguaje SFC

Las dos flechas hacia abajo indican que se avanza en la ejecución de nuevo en la etapa indicada ("Step0"); y en los bloques de acción, la 'S' (SET) simboliza que se fuerza a '1' la salida, y la 'R' (RESET) a '0'.

## 2.2 OPENPLC

Desarrollado por el profesor Thiago Alves, este software fue el primer proyecto completamente funcional y de código abierto de AP, lo cual le ha hecho ganarse mucho rédito y que se haya podido exportar a muchos y distintos hardware: Arduino, Raspberry Pi, ...

Acerca de su licencia de código abierto ([4]), que es GPL-v3.0, permite vender un producto con OpenPLC instalado. Las prohibiciones son cobrar solo por el software, bloquear el código fuente en el producto, o cambiarle el nombre al programa y venderlo como algo ajeno. Como lo que se descarga es software, la licencia no es aplicable al hardware sobre el que se cargue OpenPLC.

Para alcanzar el funcionamiento como AP, se emula uno virtual que interpreta los programas y responde o genera peticiones MODBUS/TCP ([11]). La capacidad de cargar el software en multitud de dispositivos se debe a las diferentes capas hardware que dispone OpenPLC.

OpenPLC se divide en dos subproyectos: Editor, donde el usuario diseñará sus programas de autómatas; y Runtime, la aplicación responsable de que el hardware empleado funcione como un AP. Hay un tercer subproyecto, “HMI Builder” (Human Machine Interface), el cual no va a ser empleado para nuestro proyecto por falta de tiempo, pero en un futuro puede resultar útil su explotación para la creación de interfaces con las que el usuario puede interactuar con el AP bien para un mayor control o para obtener datos durante su funcionamiento. Para obtener más información sobre él, acudir al apartado “Getting started” de la página web oficial ([3]).

### 2.2.1 Editor

Esta primera aplicación permitirá al usuario diseñar sus propios programas de autómatas pudiendo emplear cualquiera de los cinco lenguajes normalizados (punto 2.1). En la referencia [12] se adjunta su paquete de instalación y cómo cargarlo correctamente en el PC.

Un ejemplo de cómo diseñar un diagrama Ladder desde cero (cómo declarar las variables, cómo crear un contacto o una bobina, dónde se define el tiempo de ejecución del AP, ...) se encuentra paso a paso explicado en esta sección de la página web oficial de OpenPLC [13].

Complementando esta referencia, se añade en la Figura 9 lo que sería el archivo .st final que más tarde se cargará en el Runtime para ejecutar el AP virtual.

```
1 PROGRAM My_First_Program
2   VAR
3     PB1 AT %IX0.0 : BOOL;
4     PB2 AT %IX0.1 : BOOL;
5     LED AT %QX0.0 : BOOL;
6   END_VAR
7
8   LED := NOT(PB2) AND (LED OR PB1);
9 END_PROGRAM
10
11
12 CONFIGURATION Config0
13
14   RESOURCE Res0 ON PLC
15     TASK My_Task(INTERVAL := T#50ms,PRIORITY := 0);
16     PROGRAM My_Instance WITH My_Task : My_First_Program;
17   END_RESOURCE
18 END_CONFIGURATION
19
```

Figura 9: Ejemplo de un archivo de formato .st

El formato de este fichero se corresponde con el lenguaje ST (apartado 2.1.4), añadiendo al final todo lo relativo a la configuración del PLC (intervalo de ejecución de la tarea, asociar esa tarea al título del programa, ...).

### 2.2.2 Runtime

Aplicación que debe estar instalada en el dispositivo que alojará el AP y en el que se ejecutarán los programas. Como donde correrá finalmente el AP en el proyecto será desde un PC con sistema operativo Windows, esta explicación se centrará en esta opción, cuyo paquete de instalación y las instrucciones se adjuntan en el link [14] entrando en “Soft-PLC->Windows”.

La subida de los programas por parte del usuario es posible gracias a la aplicación NodeJS (más información sobre ella en su página web oficial, [15]) que es la que crea el servidor web en el puerto 8080 del PC, accesible desde cualquier explorador de Internet buscando “localhost:8080”, en el cual se puede hacer correr la aplicación y así conectarte ya al AP virtual.

Desde este servidor web, se pueden configurar todos los parámetros en los que se producirá la ejecución. Como el AP virtual se ejecutará en PC, hay que disponer de “dispositivos esclavo” (“Slave Devices”) que expandirán las I/O declaradas en el programa de AP con el Editor. Por tanto, adicionalmente a todo esto se cargará un firmware en los elementos hardware que den sus pines externos para esta tarea. Los admitidos por OpenPLC y sus desarrollos correspondientes están en la misma referencia que los paquetes de instalación ([14]).

Partiendo de que estos “Slave devices” se tomarán en este TFG como placas “Arduino UNO”, su configuración en Runtime se encuentra detallada en esta sección de la web de OpenPLC: [16].

Por último, el proceso de cómo cargar el programa de autómatas que se encuentra en el archivo .st generado al final del apartado anterior está descrito en este link de la bibliografía [17].

### 2.2.3 Alternativas a OpenPLC

Para la aplicación con la que implementar un AP en dispositivos había otros proyectos similares a OpenPLC que también eran de código libre, con la capacidad de poder diseñar un programa de autómatas y de hacer correr un AP tanto virtualmente como en ciertos uC.

- StxLadder, aplicación cuyo soporte está en castellano, combina los diagramas LD con un buen entorno gráfico, usa un lenguaje AP escrito no convencional como es Pawn. [18]
- ClassicLadder, similar a OpenPLC en muchos aspectos, también tiene una comunicación por MODBUS para intercambiar la información entre la CPU y las I/O. [19]
- OpenGrafcet, lo más destacable de esta última opción, a diferencia de las dos anteriores, es el empleo del lenguaje SFC para diseñar y simular los programas. [20]

Las razones por las que se optó por OpenPLC sobre los otros proyectos son: la versatilidad respecto al soporte en distintas arquitecturas de uC (destacando poder cargarse en un Arduino UNO, compatible con el software de los kits de Microchip); unas interfaces gráficas sencillas y claras desde el inicio; e, importante, la opción con OpenPLC para el usuario de diseñar su programa de autómatas en cualquiera de los cinco lenguajes normalizados actualmente ([capítulo 2.1.1](#)), existiendo la oportunidad de combinarse entre ellos para una misma aplicación.

## 2.3 POWER LINE COMMUNICATIONS (PLC)

Esta tecnología de comunicación por RE transmite información a través de un medio (un cable conductor) concebido originalmente para únicamente transportar potencia eléctrica.

Implementable tanto en redes AC como DC ([Tabla 1](#)), permite un ahorro considerable en costes de cableado en muchos campos de servicio habituales. Entre las opciones que ofrece Microchip en PLC (la mayoría del hardware utilizado se obtiene de aquí), hay una gran variedad de dispositivos, todos con el correspondiente software implementado, aptos tanto para aplicaciones con conexiones sencillas como soluciones que sí requieran una red de trabajo más compleja. Además, permiten operar independientemente del protocolo de comunicación a seguir o del ancho de banda en el que se trabaje ([sección 2.3.1](#)). [21].

Como ejemplos de aplicaciones cotidianas, uno de ellos es el sector de los contadores de luz. En estos aparatos, resulta muy útil el PLC para realizar la lectura remota de las mediciones clásicas de la energía y potencia eléctrica en una instalación y/o vivienda. Otros ámbitos en la sociedad actual que se apoyan en esta tecnología son los sistemas de alarmas de incendio, el alumbrado de las calles en una localidad o en las estaciones de carga de los coches híbridos.



### 2.3.1 Normativas y frecuencias

#### 2.3.1.1 Estándar G3-PLC

Normalizado según la recomendación ITU-T G9903 establecida por la Unión Internacional de Telecomunicaciones ([22]) relativa, dentro la serie G que abarca los sistemas de transmisión de señales en redes de trabajo, a “dispositivos de comunicación de banda estrecha por RE”.

Estándar, libre y normalizado a nivel internacional, promovido por la organización Alliance que aporta grandes beneficios dentro del mundo de la comunicación PLC: mayor rango, más fiabilidad y una velocidad elevada pese a los posibles entornos en los que se desarrollan estas conexiones. Así se aumenta la demanda de esta tecnología en aplicaciones industriales por su capacidad de control y monitorización a lo largo de toda la red gracias al doble sentido del flujo eléctrico (este aspecto en el ámbito tecnológico es denominado “Smart Energy”).[23]

Para implementar en un dispositivo una aplicación con esta normativa, se deberá cargar el ‘stack’ correspondiente: un ‘stack’ es el conjunto de librerías de código y módulos que definen la estructura software y la configuración en aspectos importantes como la arquitectura de los niveles y los enlaces con los programas para un correcto funcionamiento entre estos componentes. Microchip proporciona vía Internet su stack de PLC correspondiente a este protocolo, que incluye la guía del usuario.[24].

Aunque para este proyecto se escoja únicamente por trabajar con este estándar, se añade también una explicación sobre la otra alternativa existente de protocolo para comunicación por la RE: el estándar PRIME.

#### 2.3.1.2 Estándar PRIME

En este caso, la norma ITU sobre la que se ampara este protocolo es la ITU-T G9904 ([25]). También fomentado por Alliance, se muestra la aparición de PRIME (el significado completo de este acrónimo, en el [Tabla 1](#)) como una solución gratuita, abierta y libre para la utilización de la comunicación por RE que supone un gran avance en la evolución de esta tecnología y sus aplicaciones en el campo de “Smart Energy”. [26]

También será necesario aquí un ‘stack’ para ser empleado en nuestro hardware, se adjunta su referencia en la web de Microchip: [27].

Los estándares G3-PLC y PRIME son, actualmente, los que tienen el nivel más alto de desarrollo alcanzado dentro de la tecnología PLC, por lo que será habitual que en documentos técnicos se les denomine protocolos “state-of-the-art”.

#### 2.3.1.3 Capas de comunicación

Cuando un dispositivo conectado a una red de comunicación recibe o transmite información, ésta tiene que seguir un camino en él durante el que tiene que adaptarse para poder ser procesada óptimamente tanto a nivel de aplicación (en una interfaz de programación) como a nivel de físico o de hardware (en la RE). Ese recorrido tiene su modelo de referencia, gracias al “modelo de interconexión de sistemas abiertos” (o “Modelo OSI” [28]), en una división en capas de comunicación o también denominadas niveles de abstracción. En PLC, estos niveles sufren ligeras modificaciones según el protocolo, G3-PLC o PRIME, en el que se encuentren operando los elementos de la red las cuales se pueden observar ilustradas en ambos casos en la Figura 10.

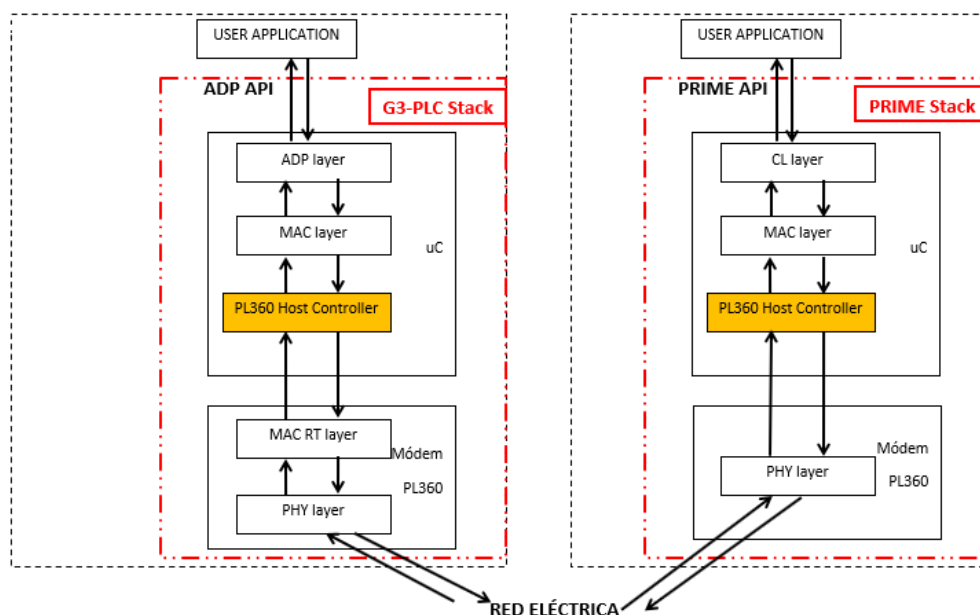


Figura 10: Estructura de las capas de comunicación según el protocolo escogido

La mayor diferencia es la existencia en G3-PLC, dentro del Módem PL360 ([subsección 2.3.2.2](#)), de una capa adicional denominada “MAC RT” (significado completo en el [Tabla 1](#)). Este bloque permite controlar desde este componente las tareas que demanden un bajo retardo de ejecución (funcionamiento en tiempo real), como la comprobación de errores que posea la aplicación con los mensajes, lo que lleva a una reducción de los tiempos del uC anfitrión. Para investigar más sobre la función concreta de las otras capas, se puede acudir a las guías de usuario de cada uno de los dos ‘stacks’: [\[24\]](#) [\[27\]](#).

En el PL360 Host Controller (nivel destacado en amarillo en los dos estándares) se encuentra la subcapa de “abstracción de hardware” (HAL) que permite la correcta compatibilidad e intercambio de información entre el uC anfitrión y el Módem PL360, ya en contacto con la capa física donde se producirá la comunicación. Así se permite al uC controlar el PL360 y acceder a la configuración del PLC desde el nivel de aplicación. Para más información, se adjunta la ficha de datos sobre la interfaz de usuario de este controlador: [\[29\]](#).

Para cerrar esta sección sobre el par de protocolos PLC, se adjunta un documento ([\[30\]](#)) con un estudio práctico de la comparación entre ellos. Resaltar un par de las diferencias más significativas, ambas en favor de G3-PLC: tiene una menor distorsión por ruido en la línea eléctrica; y posee un factor de corrección de error en los mensajes más desarrollado que PRIME.

#### 2.3.1.4 Bandas de frecuencias

En la Figura 11, extraída del documento [\[31\]](#), se representan los 3 rangos para la comunicación por RE permitidos y normalizados según el lugar donde se encuentre la aplicación (Estados Unidos, Japón o Europa), ninguno de los cuales supera el límite máximo permitido que serían los 500 kHz.

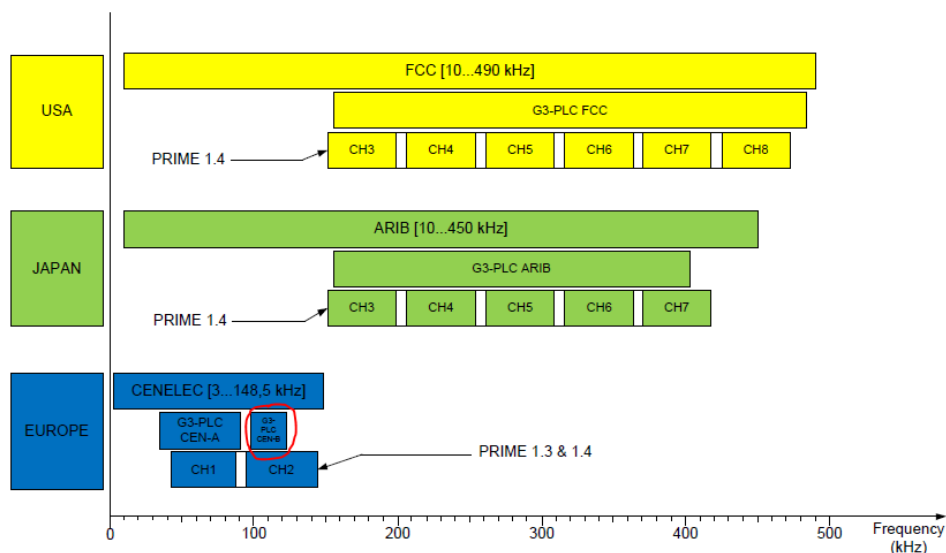


Figura 11: Bandas de frecuencias normalizadas para PLC

#### 2.3.1.4.1 Banda CENELEC-B

La norma EN 50065 sobre transmisiones por la RE de baja tensión establece que en el territorio europeo este tipo de comunicaciones se deben producir dentro del espectro de frecuencias entre 3 y 148.5 kHz ([32]). La banda G3-PLC CENELEC B, en rojo en la Figura 11, cumple con estos márgenes (CENELEC), y se mueve entre un mínimo de 95 kHz y un máximo de 125 kHz.

### 2.3.2 Hardware/dispositivos

#### 2.3.2.1 Kit de evaluación PL360G55CB-EK

Compuesto de los elementos para desarrollar un sistema de comunicación eficiente basado en PLC, esta plataforma puede utilizarse como punto de partida de aplicaciones modulares con redes de trabajo inteligentes avanzadas.

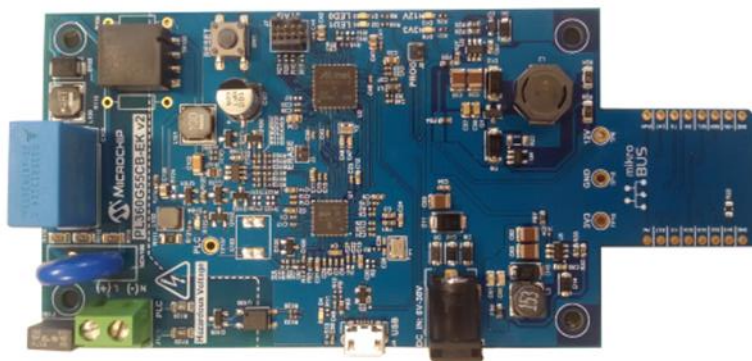


Figura 12: Vista cenital del kit PL360G55CB-EK

El nombre del modelo son acrónimos de los componentes principales del hardware: PL360, que permite “traducir” e intercambiar información entre la RE y la aplicación; G55 procede del tipo de uC principal, SAMG55J19; CB simboliza el rango de frecuencias en el que se desarrolla la comunicación PLC (CENELEC B); y EK proviene de “Evaluation Kit”.

Tiene la posibilidad de poder ser tanto alimentado eléctricamente como programado a nivel software por un mismo conector Micro-B USB, y posee un módulo llamado “mikroBUS” que proporciona pines externos para interactuar con otras PCB y así añadir nuevas funciones al kit.

La información y la imagen han sido extraídas de la Guía del Usuario (“User Guide”) oficial de Microchip, a la que se puede acceder a través del siguiente enlace: [\[31\]](#).

#### **2.3.2.2 Dispositivo PL360**

PL360 es la denominación que recibe este módem programable fundamental en los kits y PCB’s de Microchip con capacidad de originar una red PLC.

Con la versatilidad de poder operar siguiendo cualquiera de los dos protocolos de trabajo ([2.3.1.1](#)), ha sido diseñado para ser controlado por los uC anfitriones (principales) de sus kits correspondientes, lo que posibilita que el usuario a través de sus aplicaciones pueda configurar aspectos relacionados con la comunicación PLC a nivel físico también. Esta información ha sido extraída de la ficha de datos (“Datasheet”) redactada para este elemento que Microchip proporciona libremente a través de este enlace: [\[33\]](#).

En el kit PL360G55CB-EK representado en la Figura 12, de los dos chips que se observan, el correspondiente a PL360 sería el de abajo y el uC SAMG55J19 (el anfitrión) el de arriba, siendo así perfectamente integrable con el resto de los componentes de la PCB.

## 3 DESARROLLO DEL PROYECTO

### 3.1 PLANTEAMIENTO INICIAL

Siguiendo la metodología marcada en la Introducción ([apartado 1.3](#)) para alcanzar los objetivos propuestos en este TFG, el siguiente capítulo relata cómo se afrontan todos esos desafíos y qué ventajas dan para avanzar correctamente durante el desarrollo del proyecto.

En primer lugar, se elige el hardware que se emplea como AP: para conseguir la distribución de sus módulos externos por la RE mediante tecnología PLC, tienen que ser dispositivos compatibles con este tipo de comunicación, que posean un periférico con acceso a pines que se conecten a las I/O, y que admitan cargar OpenPLC en su uC. Se opta, aprovechando el enlace con Microchip, por el kit de evaluación PL360G55CB-EK (presentación en [2.3.2.1](#)) que ofrece esta empresa y que cumple con los requisitos nombrados. Además, a pesar de que sólo opere en el rango de frecuencias CENELEC-B ([subsección 2.3.1.4.1](#)), al integrar el Módem PL360 permite trabajar siguiendo cualquiera de los dos protocolos normalizados ([punto 2.3.2.2](#)).

Estas PCB necesitan estar siempre alimentadas eléctricamente, para lo que aquí se emplea el conector Micro-B USB unido por cable a un PC ya que, además, es útil para dos tareas más: cargar las aplicaciones desarrolladas en el uC correspondiente; y ‘debuggear’ (depurar) mediante mensajes por pantalla de los puertos serie asignados a cada conexión/placa/AP, así se evalúan mejor los programas y se encuentran los posibles errores más rápidamente (fundamental para los tests previos). Para el último aspecto, debe estar instalado en el PC un emulador de terminal con el que se puedan abrir los distintos puertos serie y que permita visualizar lo extraído de cada uno por separado. El utilizado será “TeraTerm”, y como se observa en la Figura 13, se pueden monitorizar los resultados de varios puertos serie al mismo tiempo.

[34]

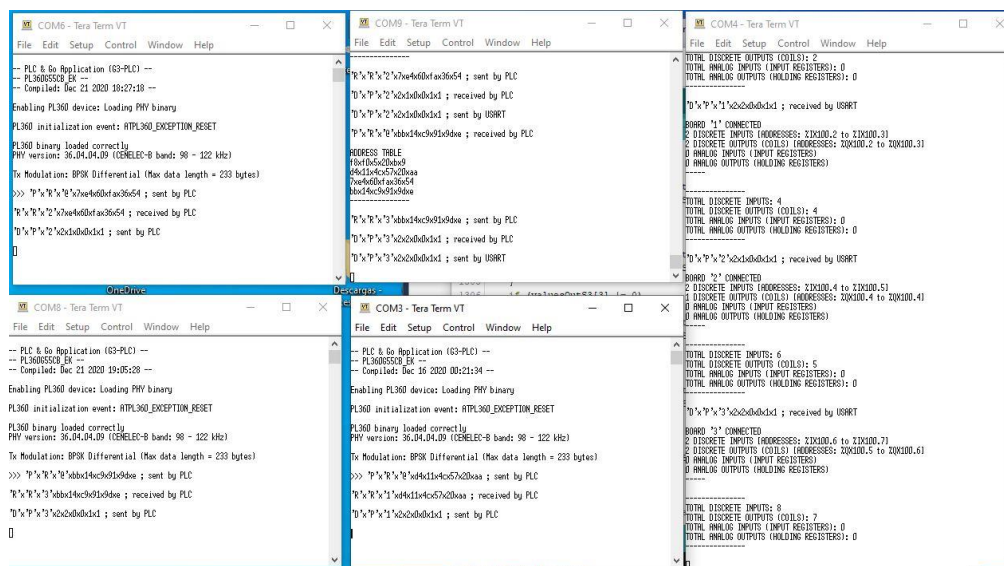


Figura 13: Ejemplo del programa TeraTerm

Aunque la intención final es hacer funcionar OpenPLC de forma independiente en cada una de estas plataformas, su investigación para comprender su modo de operación comienza con la instalación de Runtime en un PC con Windows como sistema operativo, por tanto, en esta fase el AP virtual se ejecutará en el ordenador ([subcapítulo 2.2.2](#)). Como dispositivo esclavo para

aportar las I/O se emplea un “Arduino UNO” estándar (soporte en [\[35\]](#)) en el que se debe cargar el firmware correspondiente de la página web. De esta manera, se explora también sobre la forma en la que se actualizan los diferentes registros de entradas y de salidas durante la ejecución del AP en el interior de este dispositivo.

Para la subida de aplicaciones en sus kits y placas, Microchip proporciona su propia herramienta software llamada Atmel Studio (actualmente Microchip Studio, pero con la que he estado trabajando durante toda la elaboración es la primera), un entorno de desarrollo de instalación libre en el que el usuario es capaz de escribir y compilar código en C y C++ para diseñar sus programas. [\[36\]](#)

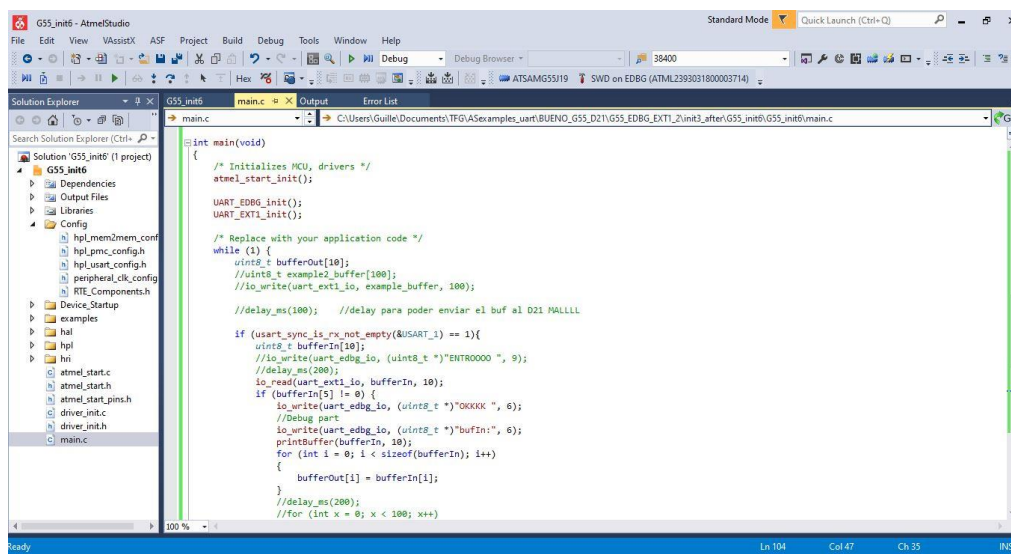


Figura 14: Vista general de la interfaz de Atmel Studio

Buscando una disposición sencilla de los módulos de I/O de cada AP, la estructura sobre la que se han apoyado durante este trabajo han sido placas protoboard, ya que tanto para las fases de test como para la presentación final permiten diferenciar con facilidad el módulo correspondiente a un AP de otro distinto, y unir mediante cables los pines externos proporcionados por cada kit PL360G55CB-EK ([punto 2.3.2.1](#)) con una entrada (por ejemplo, un pulsador) o una salida (por ejemplo, un motor) introduciéndolos en los agujeros (o ‘topos’) de la columna eléctrica adecuada. Más adelante en la Figura 16, se puede observar cómo se integra la estructura de placas protoboard al sistema, con las I/O organizadas en ella.

### 3.2 DIFICULTADES ENCONTRADAS

Aunque inicialmente se contempló que la versatilidad de OpenPLC respecto a su funcionamiento en distintas arquitecturas de hardware permitiría su implementación en el dispositivo escogido para esta tarea, el uC anfitrión de PL360G55CB-EK, en el comienzo de la investigación se descubrió que las plataformas en las que corría la aplicación tenían Linux como entorno de ejecución. Pese a ser posible lograr la compatibilidad con el sistema operativo del hardware de este proyecto (“FreeRTOS”, sistema operativo en tiempo real libre para uC, [\[37\]](#)), esta circunstancia suponía que el trabajo para adaptar el programa hubiese sido demasiado costoso de tiempo para un TFG, por tanto se tuvo que desear este procedimiento.

Este inconveniente es el más grave porque impide uno de los objetivos principales marcados en este documento: configurar cada uno de los dispositivos para funcionar como un AP independiente.

Entonces, se opta por una primera solución: demostrado en pruebas anteriores el correcto funcionamiento de esta aplicación desde un PC Windows y con un Arduino UNO como dispositivo que aporta sus pines externos como nuevos registros del sistema, la nueva meta fue que estos dispositivos “esclavo” pasasen a ser las placas de la red PLC, convirtiéndose así de controladores autónomos a ser los módulos de I/O distribuidos por la RE, una situación donde sólo habría un único AP en ejecución, el del ordenador desde OpenPLC Runtime.

Esto significaba ampliar las investigaciones: por ejemplo, había que buscar una forma en la que intercambiar la información entre el PC y la red relativa a la actualización cíclica de los registros de I/O del autómatas, o también cómo lograr subir el firmware proporcionado por OpenPLC, desarrollado originalmente para Arduino UNO, a los kits con uC SAMG55. Este último aspecto acabaría siendo primordial, ya que en principio Atmel Studio da la opción de importar proyectos en soporte Arduino y convertirlos a proyectos compilados en C++ para ser “cargables” en PCB’s de Microchip. Sin embargo, y tras multitud de intentos, el resultado en este caso volvió a ser negativo, lo que llevó a descartar esta nueva solución y produjo una reflexión profunda sobre cuál iba a ser el siguiente paso que dar en este TFG. Otro problema, aunque menos influyente, fue la falta de espacio para distribuir un número considerable de AP’s, con sus módulos de I/O incluidos, en varias placas protoboard de tamaño medio, añadiendo un espacio de separación entre ellos para que visualmente se puedan diferenciar individualmente.

Ya en el final del desarrollo del proyecto, por falta de tiempo ya que se ha priorizado asegurar un funcionamiento óptimo en el aspecto de la comunicación y en la ejecución de los AP, los módulos de I/O son sólo digitales, cuyo tratamiento de los valores binarios es más sencillo que si fueran analógicos.

### 3.3 SOLUCIÓN ADOPTADA

Estos problemas acabaron teniendo una gran influencia sobre la idea inicial propuesta para este proyecto, una de ellas no poder cumplir uno de los objetivos principales marcados al comienzo del TFG. Sin embargo, la búsqueda del alcance real seguía siendo factible: lograr una distribución por la RE mediante tecnología PLC de los módulos de expansión de I/O de un AP. De este modo, el proceso de elaborar una solución alternativa se va a describir a partir de este punto como la realización de una “Prueba de concepto” del producto final a realizar.

Para arreglar el tema de la reducción de espacio y también para obtener una visualización más clara de la estructura física del sistema, se optó por la bajada de la cantidad de módulos de I/O final en este proyecto a cuatro, número suficiente para observar unos resultados válidos para apoyar la nueva hipótesis.

A estas cuatro placas, habrá que sumar una más, también proporcionada por Microchip, ajena a la red PLC y cuyo modelo y arquitectura de uC será diferente al del kit PL360G55CB-EK. Como se analizará en el [capítulo 3.3.2](#), este dispositivo será capaz de admitir el firmware de OpenPLC con formato Arduino y así poderse comunicar con el PC e interpretar las órdenes del AP corriendo en él, hecho trascendental que hará que se pueda continuar con la elaboración del proyecto siguiendo esta nueva vía.

#### 3.3.1 Aplicación ‘PLC and Go’

Previamente a la toma de la decisión de elaborar la “Prueba de Concepto”, se realizaron los experimentos correspondientes a las pruebas de comunicación PLC entre exclusivamente los kits PL360G55CB-EK. Con el stack asignado al protocolo G3-PLC cargado en el uC, se eligió por



trabajar con un programa denominado “PLC and Go”, que originalmente ejecutaba una aplicación sencilla entre varias placas y finalmente se acabó convirtiendo en el punto de partida del desarrollo final.

Esta aplicación, incluida dentro de los ‘stacks’ de G3-PLC y PRIME de Microchip ([24]; [27]), será el desarrollo original sobre el que partirá el proceso de elaboración a nivel de software de este TFG en cada uno de los kits PL360G55CB-EK.

Estudiando la nota de aplicación (“Application Note”) facilitada por Microchip ([38]) se extrae que inicialmente ‘PLC and Go’ hace correr una aplicación en modo ‘chat’ entre varios nodos dentro de una red sencilla con tecnología PLC, de forma que se pueden intercambiar mensajes entre todos ellos siguiendo un modelo “multi-punto” de comunicación.

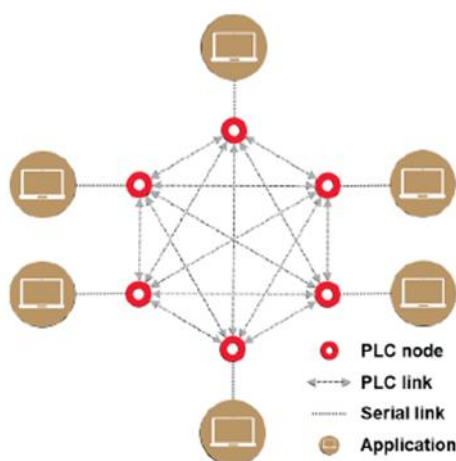


Figura 15: Representación de la comunicación PLC “multi-punto”

Los nodos se tratarán de PC’s conectados vía USB a las PCB de Microchip con el módem PL360 implementado en los que se podrá configurar las especificaciones físicas fácilmente ya que en este programa se desestiman los aspectos avanzados que implican el funcionamiento de las capas MAC de ambos protocolos (estructura en [sección 2.3.1.3](#)), no requeridas para una aplicación sencilla en un esquema de comunicación como el de la Figura 15.

La elección de este programa para ser la base para los nuevos desarrollos introducidos por el autor es por aprovechar las funciones ya creadas para el modo ‘chat’ relativas a la transmisión y recepción de mensajes por PLC, añadir el tratamiento de los pines para manejar las I/O, y estos dos aspectos combinarlos de manera óptima para lograr el resultado deseado.

### 3.3.2 Inserción de una “placa de control”

Como ya se ha introducido al principio de este capítulo, fruto de la investigación para solventar la incompatibilidad del firmware de OpenPLC con formato Arduino en el uC SAMG55 de los kits, finalmente se encontró la forma de poder configurar un nuevo dispositivo, también de Microchip, para que sí admita esa parte de la aplicación. Se localizó por la web un post en la página GitHub subido por la propia marca Microchip (en la referencia se nombra como Atmel, el antiguo nombre de la compañía) dónde se describe cómo configurar paso por paso el modelo de placa “SAMD21 Xplained Pro” de forma que, al finalizar el proceso, se pudiese controlar como un dispositivo Arduino. [39]



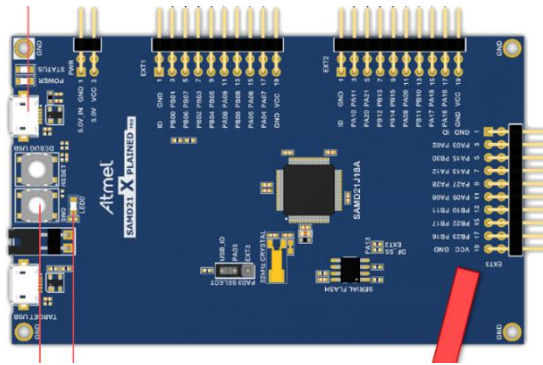


Figura 16: Vista de una placa SAMD21 Xplained Pro

Aplicándolo en este proyecto, tanto en el proceso de configuración como en la subida del firmware OpenPLC posterior el resultado en ambos casos fue satisfactorio, por tanto, se logró tras una gran cantidad de intentos la implantación de esta aplicación en una plataforma compatible con el hardware utilizado en la red PLC.

Para esta placa de control, se hace uso de los dos conectores Micro-B USB que tiene para comunicarse ambos con el PC: el denominado como “TARGET USB” se emplea para el intercambio de información vía MODBUS con la aplicación OpenPLC Runtime sobre el estado de los registros del AP virtual durante su ejecución. El otro conector, “DEBUG USB”, será útil para depurar a través de él mensajes por la pantalla del puerto serie asignado del PC y así evaluar el programa y monitorizar sus resultados.

Si se quisiera obtener información adicional del funcionamiento y estructura de esta placa al margen del que se le da para esta tarea, se adjunta la guía de usuario proporcionada por Microchip: [\[40\]](#).

### 3.3.3 Placa base y placas servicio de la red

Hablando ya de las cuatro placas que están dispuestas por la RE, en vez de ejecutar cada una por separado el programa de autómatas, en esta “Prueba de Concepto” actúan finalmente como los módulos de I/O cuya su única función es enviar información por la RE si alguna de las entradas que poseen ha variado; y gestionar los mensajes recibidos correctamente, ya que se tratan de las actualizaciones llegadas desde el PC de los registros de salidas por parte del AP virtual y es posible que fueren alguna variación en los valores de los pines dependiendo del programa.

Según lo expuesto en la sección anterior, la información que reciba la placa de control SAMD21 desde el PC es necesario que uno de los módulos de red esté conectado a ella para que la vaya intercambiando durante el transcurso del funcionamiento del sistema. Esta placa, por tanto, deberá llevar un desarrollo software adicional a las demás por este hecho, ya que ahora es la que maneja más cantidad de mensajes. Es por ello que se le va a designar como la “placa base” del sistema mientras que, al resto, siguiendo la denominación usada en comunicación PLC, van a ser las “placas servicio”, que sí que operarán puramente como módulos de expansión de I/O.

#### 3.3.3.1 Placa base

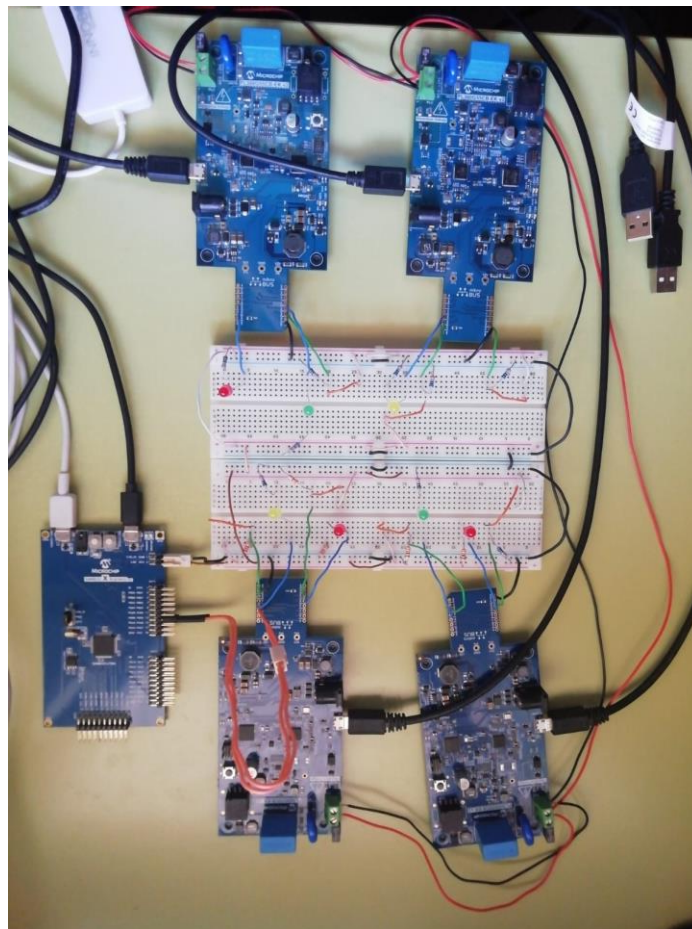
Siendo una plataforma PL360G55CB-EK como el resto de los módulos de la red PLC, en este punto se expondrán los aspectos que diferenciarán este dispositivo de los demás del mismo modelo. Esta placa base destaca sobre las servicio en:

- Ella es la conectada vía USART con la placa de control, por tanto, maneja todos los mensajes procedentes del PC relativos a la actualización de los registros de salidas del AP virtual.
- Recibe la información de las posibles variaciones en las entradas de cualquiera del resto de módulos de I/O por PLC y la envía por MODBUS de nuevo al ordenador.
- Funcionalidades adicionales, como almacenar una lista de registro de los módulos que se van uniendo a la red, útil en caso de desconexión de alguno de ellos para no repetir información.

En la temática de la gestión de I/O, el desarrollo será el mismo que para las placas servicio porque además se utilizarán pines similares para las entradas y las salidas en los cuatro dispositivos.

### 3.3.4 Análisis de los resultados obtenidos

Partiendo del necesario uso de AP's como controladores del sistema, y con la implementación del software OpenPLC y de la tecnología de comunicación PLC, pese a que finalmente en este TFG se desarrolla una Prueba de Concepto de la idea inicial planteada del producto, analizando el resultado se aprecia que se alcanza una óptima sinergia entre estos tres elementos que acaba con un mismo programa de autómatas diseñado en cualquiera de los cinco lenguajes normalizados, ejecutándose a través de cuatro módulos de expansión, cada uno con sus I/O, con un retardo casi inapreciable, y conectados entre ellos por únicamente un par de cables (al estar trabajando en tensión continua, DC, el cable rojo es el correspondiente a voltaje positivo; y el negro al negativo).



*Figura 17: Vista cenital de la estructura final*

Aunque en el sistema se ha ejecutado el AP virtual en un PC y no en cada uno de los kits como era la intención inicial, los resultados muestran que la distribución de los módulos de I/O a través de PLC se produce satisfactoriamente y con poco retardo en la transmisión de la información.

Respecto a una observación puramente del ámbito software, la implementación en los kits del software “PLC and Go” ha sido un acierto, ya que esta aplicación, continuando con la justificación dada en el [punto 3.3.1](#), opera justo encima de lo que sería la capa física del dispositivo, casi en contacto con la red PLC, lo que produce el hecho de que se pueda configurar casi por completo cómo se va a producir la comunicación en puntos como la velocidad o los perfiles de frecuencias disponibles para emplear.

Sobre el número máximo de módulos de I/O que puede manejar este sistema, aunque en la estructura final construida sea de cuatro dispositivos como ya se ha aclarado anteriormente, el software desarrollado por el autor permite conectar hasta 36 placas PL360G55CB vía PLC por la asignación interna a cada una de un carácter de dirección de ‘0’ a ‘9’ o de ‘A’ a ‘Z’.

Como se puede observar en la Figura 17, cada placa está conectada vía USB mínimo por un cable hacia el PC, lo que provoca la falta de puertos de este tipo en el ordenador para esta tarea. Es necesario obtener uno o varios “hub” USB para que amplíen el número de conectores disponibles y así solucionarlo.

Volviendo a la Figura 17, se aprecia que para la comunicación USART entre la placa de control SAMD21 y la placa base de la red únicamente se emplean también un par de cables, como para PLC. Esta vez, la función será conectar el pin Tx (transmisión) de una con el Rx (recepción) de la otra, y viceversa.

Para todos los módulos de expansión, las entradas digitales serán pulsadores manuales (diseñados con sólo un cable); y las salidas digitales diodos LED (Diodos Emisores de Luz, [Tabla 1](#)). En la configuración de los pulsadores, se debe establecer una tensión continua de alimentación común a todos, 5V extraídos del conector VCC de la placa de control, porque ésta será la que se lleve al pin correspondiente del kit cuando el pulsador se cierre para forzar el valor binario ‘1’. En esta línea, también hay que remarcar que será necesario unir las tomas de tierra de las placas, remarcadas en ellas como GND (“Ground”), en un mismo punto porque todas necesitan tener la misma referencia de tensión cero para que se produzca correctamente la comunicación vía PLC y también vía USART entre el SAMD21 y la placa base.

En la [sección 3.2](#) se introduce el problema surgido de la no implantación de elementos de campo analógicos en los módulos de la red al no disponer de suficiente tiempo para testear con ellos en las pruebas. Sin embargo, el autor incluye en el desarrollo software de la placa de control dos funciones de tratamiento de posibles valores analógicos si los hubiera de forma que, si alguien en un futuro quisiera continuar con esta investigación, puede disponer de un punto de partida para su programación y añadirla también a las placas de la red.

Dentro también de los inconvenientes ocurridos durante la elaboración, ya se ha especificado que la tarea de la adaptación de OpenPLC Runtime para permitir la ejecución autónoma en los uC anfitriones de los kits PL360G55CB-EK llevaba una complejidad de trabajo mayor que la estimada inicialmente. Por tanto, se propone seguir con esta investigación para estudios futuros como expansión de este proyecto y buscando cumplir con el objetivo principal planteado desde el principio de la distribución por la red PLC de un número de controladores programables, no sólo de sus módulos de I/O.

Otro campo a explorar dentro de las posibles formas de continuar con este trabajo es el empleo de otra forma de alimentación de los kits, lo que supondría remover cada cable de conexión USB de la placa al PC. Se perdería la posibilidad de debuggear el programa, pero si ya se sabe que funciona correctamente supone un ahorro de hasta cinco cables dentro de la estructura final (de la placa SAMD21 mantener el cable USB correspondiente a la comunicación MODBUS con el PC).

Como última reflexión, se añade un producto que ya se encuentra en el mercado y que va en la línea de la solución elaborada en este proyecto: sacado por la empresa inglesa Phoenix Contact, RADIOLINE es un producto que, como se aprecia en el vídeo del enlace final, hay un ejemplo en vivo por parte de la propia compañía en el que se comunica satisfactoriamente sin ningún cableado dos módulos de I/O separados por una distancia de 11 kilómetros entre ellos, lo cual refuerza la utilidad del alcance final marcado en este TFG. [\[41\]](#)

## 4 CONCLUSIONES

---

La evolución tecnológica que se lleva desarrollando a nivel global desde hace ya varias décadas hace que en la gran mayoría de ámbitos laborales no se detenga la búsqueda de nuevos métodos que impliquen una reducción de los tiempos de trabajo, mayor comodidad y flexibilidad, menor acción por parte del usuario, rebaja de costes de producción y mantenimiento, etc.

El objetivo que se perseguía con este proyecto era ofrecer una solución realizable de minimización de cableado en los entornos industriales entre los módulos de I/O y sus autómatas programables, lo que implica cumplir muchos de los aspectos enumerados en el párrafo anterior y resultar un producto interesante y rentable para compañías cuyos sistemas productivos se sitúen en naves o instalaciones de este tipo.

El sistema construido en este proyecto consigue extender por la red eléctrica hasta 36 módulos de I/O (en este TFG se han conectado cuatro) pertenecientes a un AP mediante únicamente un par de cables, en un espacio reducido y con la posibilidad de ejecutar simultáneamente el mismo programa de autómata diseñado por el usuario en el lenguaje que él desee, empleando para ello una aplicación de código libre (OpenPLC).

Todo esto da como resultado un producto que implantado en la cadena de producción de una fábrica industrial, produce una reducción de los gastos de instalación y mantenimiento, al acercar los módulos de I/O a los autómatas y aumentar la flexibilidad de estos elementos ya que se podrían mover dentro de la instalación a donde hagan falta con total facilidad con sólo sustituir los ‘buses de campo’, cuyo cableado puede estar de punta a punta de la nave, por la tecnología PLC usada en este proyecto, que permite la distribución de módulos de expansión de manera sencilla.

## 5 BIBLIOGRAFÍA

---

Como introducción de este apartado aclarar que, en esta memoria, el estilo bibliográfico que se ha seguido durante todo el documento ha sido el estilo IEEE (siglas explicadas en el [Tabla 1](#)), cuyas características más destacadas son: referencias numeradas en el orden que aparecen; una vez asignadas con un número, se usará el mismo para las siguientes veces que se cite; y que no es necesario detallar el autor ni la fecha de publicación en la referencia del documento (para saber cómo citar cada tipo de referencia que haya empleado y obtener más información, en [\[42\]](#)).

- [1 M. Rodríguez, «Revistadigital: Redes de comunicación industriales y buses de campo,» 29  
] Noviembre 2012. [En línea]. Available: <https://revistadigital.inesem.es/gestion-integrada/redes-de-comunicacion-industriales-y-buses-de-campo>. [Último acceso: 14 Enero 2021].
- [2 T. R. Alves, M. Buratto, F. M. de Souza y T. V. Rodrigues, «OpenPLC: An open source  
] alternative to automation,» 2014. [En línea]. Available: <https://ieeexplore.ieee.org/abstract/document/6970342>. [Último acceso: 12 Enero 2021].
- [3 T. R. Alves, «OpenPLC Project,» [En línea]. Available: <https://www.openplcproject.com/>.  
] [Último acceso: 13 Enero 2021].
- [4 T. R. Alves, «GitHub: Open Source License OpenPLC,» [En línea]. Available:  
] <https://github.com/thiagoralves/OpenPLC/blob/master/LICENSE>. [Último acceso: 14 Enero 2021].
- [5 «Informática básica: ¿Qué es hardware y software?,» [En línea]. Available:  
] <https://edu.gcfglobal.org/es/informatica-basica/que-es-hardware-y-software/1/>. [Último acceso: 19 Enero 2021].
- [6 B. P. Barreto Véliz, «Controladores programables,» 22 Octubre 2009. [En línea]. Available:  
] <https://www.monografias.com/trabajos75/controladores-programables/controladores-programables.shtml>. [Último acceso: 25 Enero 2021].
- [7 «Sistemas Automatas De Control De Procesos,» Julio 2015. [En línea]. Available:  
] <http://sistemasautomatasj.blogspot.com/p/blog-page.html>. [Último acceso: 24 Enero 2021].
- [8 «FBD/LD/IL Element 'Branch',» [En línea]. Available: [https://help.codesys.com/api-content/2/codesys/3.5.12.0/en/\\_cfs\\_fbd\\_ld\\_il\\_element\\_branch/](https://help.codesys.com/api-content/2/codesys/3.5.12.0/en/_cfs_fbd_ld_il_element_branch/). [Último acceso: 23 Enero 2021].
- [9 Universidad de Alcalá, «Controladores lógicos programables. Programación en lenguajes LD  
] e IL,» [En línea]. Available: <https://www.cartagena99.com/recursos/alumnos/apuntes/03-PLC-I.pdf>. [Último acceso: 24 Enero 2021].

- [1 D. Collins, «Why is the instruction list (IL) language for PLCs falling out of favor?,» 16 Julio 0] 2019. [En línea]. Available: <https://www.motioncontroltips.com/why-is-the-instruction-list-il-language-for-plcs-falling-out-of-favor/>. [Último acceso: 24 Enero 2021].
- [1 J. Bartolomé, «El protocolo MODBUS,» 2011. [En línea]. Available: 1] <http://www.tolaemon.com/docs/modbus.htm>. [Último acceso: 20 Mayo 2021].
- [1 T. R. Alves, «OpenPLC Editor,» [En línea]. Available: 2] <https://www.openplcproject.com/plcopen-editor/>. [Último acceso: 20 Febrero 2021].
- [1 T. R. Alves, «Creando tu primer proyecto con OpenPLC Editor,» [En línea]. Available: 3] <https://www.openplcproject.com/reference/basics/first-project>. [Último acceso: 11 Abril 2021].
- [1 T. R. Alves, «Instalación OpenPLC Runtime,» [En línea]. Available: 4] <https://www.openplcproject.com/runtime>. [Último acceso: 11 Abril 2021].
- [1 OpenJS Foundation, «NodeJS.org,» [En línea]. Available: <https://nodejs.org/es>. [Último 5] acceso: 11 Abril 2021].
- [1 T. R. Alves, «OpenPLC in Arduino,» [En línea]. Available: 6] <https://www.openplcproject.com/runtime/arduino/>. [Último acceso: 28 Mayo 2021].
- [1 T. R. Alves, «Uploading your project to OpenPLC,» [En línea]. Available: 7] <https://www.openplcproject.com/reference/basics/upload>. [Último acceso: 11 Abril 2021].
- [1 «StxLadder,» [En línea]. Available: <http://www.slicetex.com/ladder/>. [Último acceso: 14 8] Enero 2021].
- [1 «ClassicLadder,» [En línea]. Available: <https://sites.google.com/site/classicladder/>. [Último 9] acceso: 14 Enero 2021].
- [2 «OpenGrafcet,» [En línea]. Available: <https://sourceforge.net/projects/opengrafcet/>. 0] [Último acceso: 14 Enero 2021].
- [2 Microchip Technology, «Power Line Communications,» 2021. [En línea]. Available: 1] <https://www.microchip.com/en-us/products/smart-energy-metering/power-line-communications>. [Último acceso: 5 Mayo 2021].
- [2 ITU, «ITU-T G9903 G3-PLC,» 2020. [En línea]. Available: [https://www.itu.int/rec/T-REC-2\] G.9903/es](https://www.itu.int/rec/T-REC-2] G.9903/es). [Último acceso: 5 Mayo 2021].
- [2 Alliance, «G3-PLC Alliance,» 2019. [En línea]. Available: [https://g3-plc.com/what-is-g3-3\] plc/g3-plc-overview/](https://g3-plc.com/what-is-g3-3] plc/g3-plc-overview/). [Último acceso: 5 Mayo 2021].
- [2 Microchip Technology, «G3-PLC Firmware Stack and User Guide,» Mayo 2021. [En línea]. 4] Available: [https://www.microchip.com/SWLibraryWeb/product.aspx?product=SE\\_G3\\_PLC\\_STACK](https://www.microchip.com/SWLibraryWeb/product.aspx?product=SE_G3_PLC_STACK). [Último acceso: 14 Enero 2021].

- [2] ITU, «ITU-T G9904 PRIME,» 2013. [En línea]. Available: <https://www.itu.int/rec/T-REC-5/G.9904/es>. [Último acceso: 5 Mayo 2021].
- [2] Alliance, «PRIME Alliance,» 2020. [En línea]. Available: <https://www.prime-alliance.org/alliance/>. [Último acceso: 5 Mayo 2021].
- [2] Microchip Technology, «PRIME Firmware Stack and User Guide,» Abril 2020. [En línea].  
7] Available:  
[https://www.microchip.com/SWLibraryWeb/product.aspx?product=SE\\_PRIME\\_Stack](https://www.microchip.com/SWLibraryWeb/product.aspx?product=SE_PRIME_Stack).  
[Último acceso: 4 Mayo 2021].
- [2] Wikipedia, «Modelo OSI,» 2021. [En línea]. Available:  
8] [https://es.wikipedia.org/wiki/Modelo\\_OSI](https://es.wikipedia.org/wiki/Modelo_OSI). [Último acceso: 24 Mayo 2021].
- [2] Microchip Technology, «PL360 Host Controller,» Agosto 2020. [En línea]. Available:  
9] <https://ww1.microchip.com/downloads/en/DeviceDoc/PL360-Host-Controller-DS50002738E.pdf>. [Último acceso: 10 Mayo 2021].
- [3] M. Hoch, «Comparison of PLC G3 and PRIME,» 2011. [En línea]. Available:  
0] [https://e2e.ti.com/cfs-file/\\_\\_key/communityserver-discussions-components-files/171/6114.G3\\_2D00\\_PRIME\\_2D00\\_COMPAR.pdf](https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/171/6114.G3_2D00_PRIME_2D00_COMPAR.pdf). [Último acceso: 4 Mayo 2021].
- [3] Microchip Technology, «PL360G55CB-EK User Guide,» 2019. [En línea]. Available:  
1] <http://ww1.microchip.com/downloads/en/DeviceDoc/PL360G55CB-EK-User-Guide-DS50002871A.pdf>. [Último acceso: 10 Abril 2021].
- [3] UNE, «Norma UNE-EN 50065-1:2012,» 8 Febrero 2012. [En línea]. Available:  
2] <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0048827>. [Último acceso: 4 Mayo 2021].
- [3] Microchip Technology, «PL360 Datasheet,» Agosto 2019. [En línea]. Available:  
3] <https://ww1.microchip.com/downloads/en/DeviceDoc/PL360-Data-Sheet-DS70005364C.pdf>. [Último acceso: 10 Mayo 2021].
- [3] TeraTerm Project, «Tera Term Home Page,» 1 Julio 2020. [En línea]. Available:  
4] <https://ttssh2.osdn.jp/index.html.en>. [Último acceso: 10 Mayo 2021].
- [3] «Arduino Home Web Page,» [En línea]. Available: <https://www.arduino.cc/>. [Último acceso: 5] 18 Enero 2021].
- [3] Microchip Technology, «Microchip Studio,» [En línea]. Available:  
6] <https://www.microchip.com/en-us/development-tools-tools-and-software/microchip-studio-for-avr-and-sam-devices>. [Último acceso: 7 Mayo 2021].
- [3] FreeRTOS, «FreeRTOS,» [En línea]. Available: <https://www.freertos.org/>. [Último acceso: 11] 7] Junio 2021].
- [3] Microchip Technology, «PLC and Go Application Note,» 2020. [En línea]. Available:  
8] <http://ww1.microchip.com/downloads/en/Appnotes/AN3400-PLC-and-Go-00003400A.pdf>. [Último acceso: 10 Abril 2021].



- [3] T. Viard, «Atmel SAMD21-Xplained Pro Arduino module wiki,» 21 Marzo 2017. [En línea].  
 9] Available: <https://github.com/AtmelUniversityFrance/atmel-samd21-xpro-boardmanagermodule/wiki/GettingStarted:-SAMD21-Xplained-Pro>. [Último acceso: 11 Mayo 2021].
- [4] Microchip Technology, «SAMD21 Xplained Pro User Guide,» 10 Diciembre 2016. [En línea].  
 0] Available: [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42220-SAMD21-Xplained-Pro\\_User-Guide.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42220-SAMD21-Xplained-Pro_User-Guide.pdf). [Último acceso: 3 Junio 2021].
- [4] Phoenix Contact UK, «Quick, easy and reliable wireless I/O communication,» 21 Enero 2021.  
 1] [En línea]. Available: [https://www.rs-online.com/designspark/quick-easy-and-reliable-wireless-io-communication?cm\\_mmc=ES-EM\\_-\\_DSN\\_20210125\\_-\\_DM351936\\_-\\_TWIN\\_URL2&cid=DM351936&bid=1132535400&dtm\\_em=54ae5dc41a0f91767795ae31b458d17023c04f8a934b240da58edce8c874c270](https://www.rs-online.com/designspark/quick-easy-and-reliable-wireless-io-communication?cm_mmc=ES-EM_-_DSN_20210125_-_DM351936_-_TWIN_URL2&cid=DM351936&bid=1132535400&dtm_em=54ae5dc41a0f91767795ae31b458d17023c04f8a934b240da58edce8c874c270). [Último acceso: 25 Enero 2021].
- [4] Universidad Autónoma de Madrid, «Citas y elaboración de bibliografía: el plagio y el uso  
 2] ético de la información: Estilo IEEE,» 17 Diciembre 2020. [En línea]. Available: [https://biblioguias.uam.es/citar/estilo\\_ieee](https://biblioguias.uam.es/citar/estilo_ieee). [Último acceso: 20 Enero 2021].
- [4] G. Villafaña Samper, «Repositorio GitHub con la documentación y los ficheros fuente de este  
 3] TFG,» 21 Junio 2021. [En línea]. Available: [https://github.com/GuillermoVillaS/TFG\\_AutomatasRedPLC.git](https://github.com/GuillermoVillaS/TFG_AutomatasRedPLC.git). [Último acceso: 21 Junio 2021].
- [4] F. Vermeulen, «Guide to Create a User Manual,» 15 Febrero 2019. [En línea]. Available:  
 4] <https://instrktiv.com/en/user-manual-template/>. [Último acceso: 3 Junio 2021].

## 6 ANEXOS

---

En este TFG, el Anexo B será adjuntado como un enlace web a la página GitHub, así se puede acceder a los proyectos software y desarrollos de código realizados para cada una de las plataformas hardware que han intervenido en el proyecto. Además, para facilitar en el futuro la posible continuación de este trabajo, se administran también en la referencia las aplicaciones binarias (formato .bin) que se deben subir a los uC de los kits PL360G55CB-EK, y el proyecto en formato Arduino correspondiente al SAMD21 XPLAINED PRO. Además, también se incluirán tanto este documento de Memoria como el Anexo A que contiene el Manual de uso.

El enlace a GitHub es el siguiente: [\[43\]](#)

### 6.1 ANEXO A: MANUAL DE USO DEL SISTEMA

Se trata de un documento en formato pdf cuya intención principal es proporcionar asistencia a los usuarios que quieran utilizar este producto electrónico de una forma correcta, segura y efectiva.

Dispondrá de información tanto escrita como visual (figuras, capturas de pantalla, tablas) para ayudar al usuario con las distintas tareas de funcionamiento del sistema y proporciona la información necesaria para entender si la configuración realizada durante el proceso de trabajo ha sido la adecuada o no (corrección de posibles errores). [\[44\]](#)

### 6.2 ANEXO B: CARPETA CON LOS PROYECTOS SOFTWARE Y LAS APLICACIONES BINARIAS

En su interior contiene los ficheros correspondientes al proyecto Arduino completo desarrollado en el caso de la placa SAMD21; y los stacks de G3-PLC con la aplicación “PLC and Go” modificada en este TFG para los kits de las placas con la tecnología PLC, diferenciando según sea la placa base de la red o una de las de servicio.

Además, para agilizar la posible carga de los programas, se incluye dentro del directorio correspondiente a cada placa junto a su stack, la aplicación binaria que corre en los PL360G55CB-EK con el software implementado por el autor (la carga en el SAMD21, al hacerse desde el IDE de Arduino lleva un proceso diferente) facilitando la rápida ejecución del sistema final desarrollado en este TFG, si se quisieran hacer cambios estos archivos ya no serían útiles.